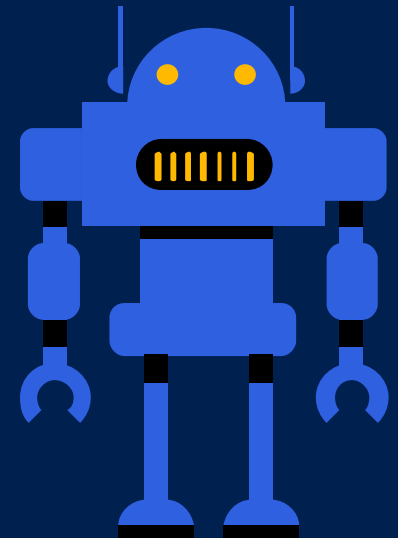
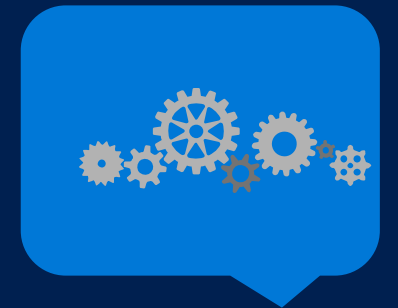


Building blocks for building bots

Bhargav Nookala, Rita Zhang
[@bhargav](#), [@ritazzhang](#)



A long history of bots

- ELIZA
- Wargames (the movie)
- SmarterChild
- IRC bots, MSN, ICQ
- Clippy

It looks like you're writing a letter.

Would you like help?

- Get help with writing the letter
- Just type the letter without help

Don't show me this tip again



Then (channels)

- Yahoo messenger
- AIM
- MSN
- IRC
- ICQ
- A few more...



Now (channels)





“Conversation as a platform”

Bots are just another form of user interface

Examples of “conversation”

- A user performs a search on any search engine
- A user is inputting medical data records, going through a form which is changing based on their input
- A user attempts to pay fines on a government website
- A user attempts to file their taxes

What are some of the problems here?

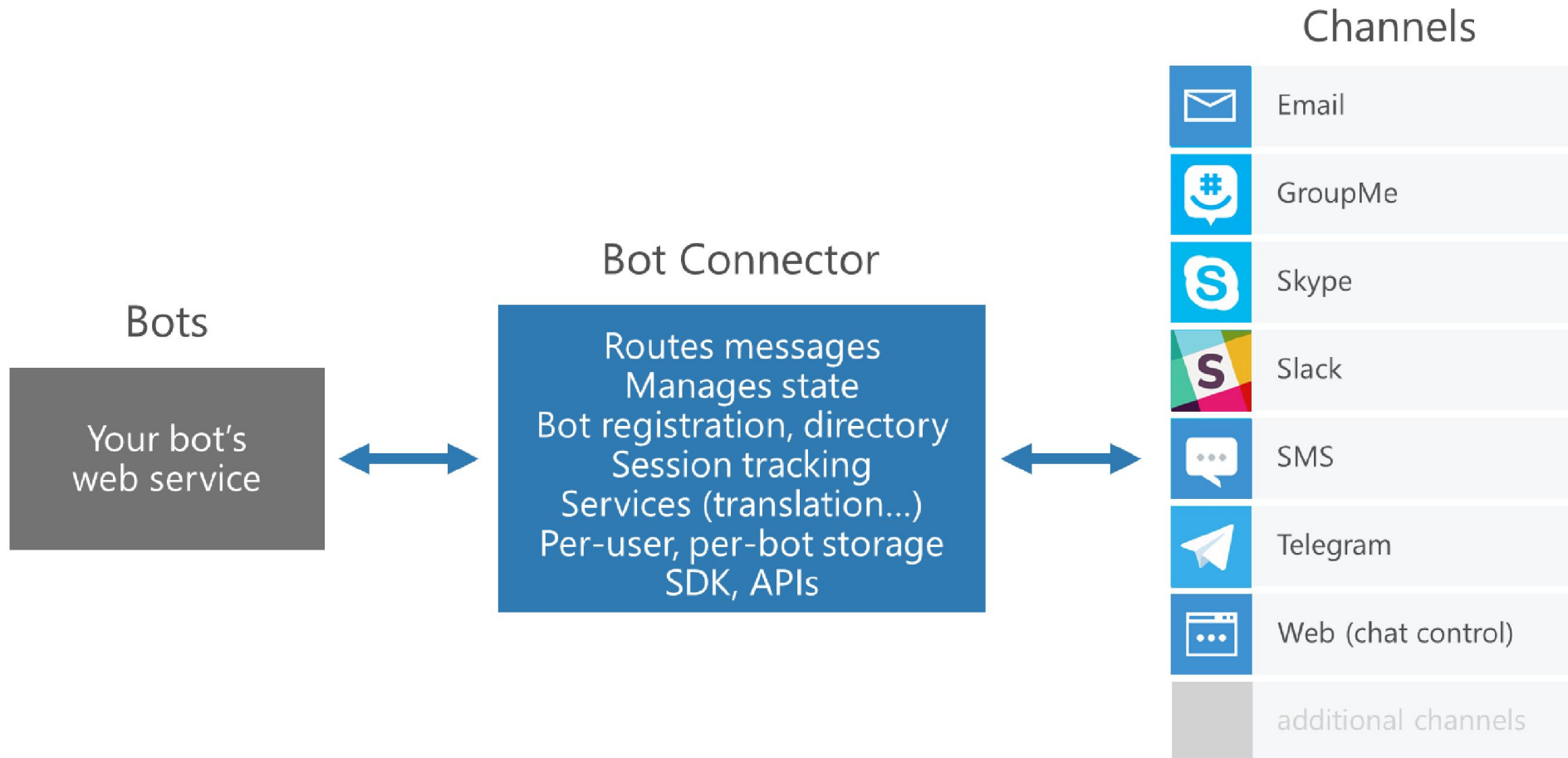
- Multiple APIs to implement
- Keeping track of user sessions/where they are in our conversation
- Understanding what users are saying/how they are interacting with your bot and responding accordingly



Bot framework joins the party

- Write once/connect to (some) things:
 - Skype, Slack, Messenger, Kik, Telegram, SMS, Office 365, and more to come...
- Features:
 - Prompts for simple things like Yes/No, Strings, Numbers, Enumerations, Attachments
 - Utilization of powerful intelligence frameworks, like LUIS
 - Built in session and state management

An Overview



OK let's create a bot!!

- Install emulator
 - Windows – requires a UI emulator.
 - Mac OS/Linux – requires Mono + console application.
- Create an App ID/App Password: <https://apps.dev.microsoft.com/>
- Node: `npm install --save botbuilder`
- NuGet: `nuget install Microsoft.Bot.Builder`

Code example:

<https://github.com/bnookala/bot-framework-examples/tree/master/hellobot>

Bots with state

- Bot Framework provides convenient wrappers around persistent state, which is backed into Azure Data Tables.
- Ex: userData, dialogData, conversationData, privateConversationData
- But, you can use whatever you want here (Redis, Mongo, etc.)
- Code example:
<https://github.com/bnookala/bot-framework-examples/tree/master/stateful>

Smarter Bots

- Make your bot smarter using various cloud based services
- Vision
- Speech
- Language
- Knowledge
- Search

Offerings: <https://www.microsoft.com/cognitive-services/>

Demo: Huma.AI

- Huma.AI – a SF based startup that is experimenting with voice commands for building interfaces
- Cognitive Service APIs used:
 - CRIS – Custom Recognition Intelligent Service for domain specific vocabulary recognition
 - LUIS – Language Understanding Intelligent Service for intent recognition
- <https://sttdemo.azurewebsites.net/>

Demo: “Finding Friends”

- Our customer wants to connect their users via photos
- Cognitive Service APIs used:
 - Face Recognition
- <https://facedetectme.azurewebsites.net>

Bot design

- Do: design bots as you were also designing a website with the same functionality or using the same API.
- Don't: overcomplicate; the uncanny valley is always going to be a problem



The wild west of smart bots

- We're kind of just scratching the surface here, and figuring things out as we go
- Bot framework is a preview for a reason
- Maybe next year we decide bots aren't cool anymore?

Thank you.

- Questions?
- Reach out to us:
 - bhnook@microsoft.com, ritazh@microsoft.com
 - [@bhargav](#), [@ritazzhang](#)



Resources

- Bot Framework – <https://dev.botframework.com/>
- Cognitive Services – <https://aka.ms/cognitive-services>
- Code Examples – <https://aka.ms/node-bot-examples>
- Facial Recognition Bot – <https://github.com/ritazh/facedetect-bot>
- Speech to Text Bot – <https://github.com/ritazh/speech-to-text-demo>
- Speech to Text case study – <https://aka.ms/huma-case-study>